

- 11                             (a) selecting a page having "free" status  
12                             and changing the status of said page  
13                             to "cacheable"; and  
14                             (b) dynamically allocating a block of  
15                             said page to be accessed by said  
16                             processor;

17

18                             if a further block of memory is required  
19                             for storage of data local to a specific processor then:

- 20                             (a) if a page having "cacheable" status  
21                             has an unallocated block, dynamically  
22                             allocating said block of said page to  
23                             be accessed by said processor; and  
24                             (b) if no block of a page having  
25                             "cacheable" status is available then  
26                             selecting a page having "free"  
27                             status, changing the status of said  
28                             page to "cacheable" and dynamically  
29                             allocating a block of said page to be  
30                             accessed by said processor;

31

32                             if a block of memory is required for  
33                             storage of data to be accessed by more than one processor  
34                             then:

- 35                             (a) selecting a page having "free" status  
36                             and changing the status of said page  
37                             to "non-cacheable"; and  
38                             (b) dynamically allocating a block of  
39                             said page to be accessed by any  
40                             processor;

41

42                             if a further block of memory is required  
43                             for storage of data to be accessed by more than one  
44                             processor then:

- 45                             (a) if a page having "non-cacheable"  
46                             status has an unallocated block,  
47                             dynamically allocating said block of

48                   said page to be accessed by any  
49                   processor; and  
50                 (b) if no block of a page having "non-  
51                 cacheable" status is available then  
52                 selecting a page having "free"  
53                 status, changing the status of said  
54                 page to "non-cacheable" and  
55                 dynamically allocating a block of  
56                 said page to be accessed by any  
57                 processor;

58                   retaining a page record as to the status of  
59                 each page; and

60                   retaining an allocation record as to which  
61                 blocks of a page have been allocated.

1                 9                 8  
1                 10. (New) A method according to claim ~~9~~  
2                 wherein if an allocated block is no longer required, the  
3                 allocation record is amended to discard the allocation of  
4                 the block.

1                 10                 8  
1                 11. (New) A method according to claim ~~9~~  
2                 wherein if no blocks on a page of memory having  
3                 "cacheable" or "non-cacheable" status are allocated, the  
4                 status of said page is changed to "free".

1                 11                 9  
1                 12. (New) A method according to claim ~~10~~  
2                 wherein if no blocks on a page of memory having  
3                 "cacheable" or "non-cacheable" status are allocated, the  
4                 status of said page is changed to "free".

1                 12                 8  
1                 13. (New) A method according to claim ~~9~~  
2                 wherein the step of discarding the allocation of a block  
3                 allocated from a page having "cacheable" status comprises  
4                 the step of discarding the data of the block.

1                 14. (New) A microprocessor system comprising:

2                   at least two processors, each processor having  
3                   a cache memory; and

4                   a system memory which is divided into pages,  
5                   each of which initially has a "free" status and is  
6                   subdivided into unallocated blocks;

7                   wherein the system is responsive to a first  
8                   request for allocation of memory space of cacheable or  
9                   non-cacheable type, by:

10                  dynamically allocating a block of memory  
11                  from a page of "free" status, the system thereafter  
12                  changing the status of said page from "free" to  
13                  "cacheable" or "non-cacheable" as the case may be; and  
14                  is responsive to a further request for  
15                  allocation of memory space of cacheable or non-cacheable  
16                  type, by:

17                  dynamically allocating a block of memory  
18                  from a page of appropriate status; or

19                  if such a block is unavailable,  
20                  dynamically allocating a block from a page having "free"  
21                  status, the system thereafter changing the status of said  
22                  page from "free" to "cacheable" or "non-cacheable" as the  
23                  case may be.

1                  ~~2~~ 15. (New) A system according to claim 14  
2                  wherein the system is responsive to a request that an  
3                  allocated block of memory is to be discarded.

1                  ~~3~~ 16. (New) A system according to claim 14, the  
2                  system further being responsive to a request to discard a  
3                  block in that if said block is the only allocated block  
4                  on the relevant page of memory then the system changes  
5                  the status of said page to "free".

1                  ~~4~~ 17. (New) A system according to claim 14  
2                  wherein the cache memory of each processor is divided  
3                  into lines.

1        5 18. (New) A system according to claim 14  
 2 wherein the size of the blocks of the system memory is a  
 3 whole multiple of the size of the lines.

1        6 19. (New) A system according to claim 15  
 2 wherein the cache memory of each processor is divided  
 3 into lines.

1        7 20. (New) a system according to claim 19  
 2 ~~wherein~~ ~~wherein~~ the size of the blocks of the system memory is a  
 3 whole multiple of the size of the lines.

2  
 6  
 VEE  
 5/22/00

#### REMARKS

A Request For One-Month Extension plus the fee  
 is enclosed.

In the office action, the Examiner has objected  
 that the priority claim in the declaration is incorrect.  
 We enclose with this response a new declaration with the  
 correct priority date inserted.

Claims 1-8 have been canceled. New claims 9  
 and 14 correspond substantially to original claims 1 and  
 3 respectively.

The Examiner objected to original claim 3 as  
 failing to comply with 35 U.S.C. § 112. It is submitted  
 that this objection is now moot in view of the  
 cancellation of original claim 3 and the insertion of new  
 claim 14.

The Examiner also objected to original claims 1  
 and 8 under 35 U.S.C. § 103(a) as being unpatentable over  
 U.S. Patent No. 4,885,680 (Anthony) in view of U.S.  
 Patent No. 5,897,660 (Reinders), U.S. Patent No.  
 5,321,834 (Weiser) and U.S. Patent No. 5,075,848 (Lai).

New claim 14 explicitly recites that the system  
 comprises a system memory which is "divided into pages,  
 each of which initially has a "free" status and is  
 subdivided unto unallocated blocks."